IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 09/739,618	§	Examiner:	Duong, Thomas
Filed: December 18, 2000	§	Group/Art Unit:	2145
Inventor(s):	§	Atty. Dkt. No:	5181-59100
John H. Howard	§		
	§		
	§		
	§		
Title: Object-Based Storage Device with Improved Reliability and Crash Recovery	§		
	§		
	§		
	§		
	§		
	§		
	§		

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed July 23, 2007, with Preappeal Brief Request for Review, and the decision thereon mailed November 23, 2007, Appellant presents this Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The present application is owned by Sun Microsystems, Inc. An assignment of the present application to the owner is recorded at Reel 011810, Frame 0381.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to Appellant.

III. STATUS OF CLAIMS

Claims 2-10, 12-20, 22-27, and 29-35 are pending. Claims 1, 11, 21, 28, and 36-44 are cancelled. Claims 2-3, 8, 12-15, 18, 22-23, 29-32, and 35 are rejected under 35 U.S.C. § 102(b). Claims 4-7, 9-10, 16-17, 19-20, 24-27, and 33-34 are rejected under 35 U.S.C. § 103(a). It is these rejections that are being appealed. A copy of claims 2-10, 12-20, 22-27, and 29-35 is included in the Claims Appendix attached hereto.

IV. STATUS OF AMENDMENTS

No unentered amendment to the claims has been filed after final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 2 is directed to a storage (12A, Fig. 2) comprising a non-volatile memory (44, 48, Fig. 2) storing a first inode locating a first file in said storage and also storing a journal (70, Fig. 4) comprising a list of committed inodes, and a block manager (42, Fig. 2). The block manager is configured to copy the first inode to a second inode (see, e.g., specification page 15, line 25-page 16, line 1), to change the second inode in response to updates to the first file (see e.g., specification page 15, line 25-page 16, line 1), and to atomically update the first file in response to a commit of the first file (see, e.g., specification page 15, line 28-page 16, line 1). The atomic update may be accomplished by writing the second inode to the non-volatile memory, whereby the

second inode locates the first file in the storage (see, e.g., specification page 16, line 25-page 17, line 5). The block manager is configured to record the second inode in the journal (see, e.g., specification page 16, line 25-page 17, line 5). See generally, e.g., Fig. 2 and Fig. 4, and specification page 9, line 26-page 10, line 14; page 15, line 14-page 17, line 24).

Independent claim 8 is directed to an apparatus comprising a computing node (10A, Fig. 2) configured to perform one or more write commands to a file and a commit command committing the one or more write commands to the file (see, e.g., specification page 13, lines 10-11 and page 14, lines 15-19), and a storage (12A, Fig. 2) coupled to receive the write commands and the commit command (see, e.g., specification page 13, lines 10-11 and page 14, lines 15-19). The storage is configured to copy one or more blocks of the file (B2, Figs. 10 and 11) to a copied one or more blocks (B2', Figs. 10 and 11), wherein the copied blocks are the blocks updated by the write commands (see, e.g., specification page 15, lines 25-26). The storage is configured to update the copied blocks with write data corresponding to the write commands (see, e.g., specification page 15, lines 25-26). Additionally, the storage is configured to copy a first inode (164, 176) locating the file in the storage to a second inode (166, 177), and to update pointers within the second inode pointing to the blocks to point to the copied blocks (see, e.g., specification page 15, line 26-page 16, line 6). The storage is configured to atomically update the file by writing the second inode responsive to the commit command (see, e.g., specification page 16, line 25-page 17, line 5). The first inode is stored in an inode file (72) that is identified by a master inode (78A), and the inode file is atomically updated with the second inode by writing the master inode subsequent to the commit command (see, e.g., specification page 16, line 25-page 17, line 5). See, generally, e.g., Figs. 2, 4, 10, and 11; specification page 9, line 26-page 10, line 14; page 15, line 14-page 17, line 24.

Independent claim 12 is directed to a method. A first inode is copied to a second inode, wherein the first inode locates a first file in a storage (12A, Fig. 2) (see, e.g., specification page 15, line 25-page 16, line 1). The second inode is modified in response

to one or more changes to the first file (see e.g., specification page 15, line 25-page 16, line 1). The first file is atomically updated by establishing the second inode as the inode for the first file (see, e.g., specification page 15, line 28-page 16, line 1). Establishing the second inode comprises storing the second inode in a journal (70, Fig. 4), wherein the journal is stored in a nonvolatile memory (see, e.g., specification page 16, line 25-page 17, line 5). See generally, e.g., Figs. 2 and 4; specification page 9, line 26-page 10, line 14; page 15, line 14-page 17, line 24.

Independent claim 22 is directed to a storage (12A, Fig. 2) comprising a non-volatile memory (44, 48, Fig. 2) storing a first inode locating a first version of a file in said storage and also storing a journal (70, Fig. 4) comprising a list of committed inodes. The storage further comprises a block manager (42) configured to copy the first inode to a second inode (see, e.g., specification page 15, line 25-page 16, line 1), to change the second inode in response to updates to the file (see, e.g., specification page 15, line 25-page 16, line 1), and to atomically update the file in response to a commit of the file(see, e.g., specification page 15, line 28-page 16, line 1). The atomic update produces a second version of the file. The atomic update may be accomplished by writing the second inode to the non-volatile memory, wherein the second inode locates the second version of the file in the storage (see, e.g., specification page 16, line 25-page 17, line 5). The block manager is configured to record the second inode in the journal (see, e.g., specification page 16, line 25-page 17, line 5). See generally, e.g., Figs. 2 and 4; specification page 9, line 26-page 10, line 14; page 15, line 14-page 17, line 24.

Independent claim 29 is directed to a method. A first inode is copied to a second inode, wherein the first inode locates a first version of a file in a storage (12A, Fig. 2) (see, e.g., specification page 15, line 25-page 16, line 1). The second inode is modified in response to one or more changes to the file, creating a second version of the file (see, e.g., specification page 15, line 25-page 16, line 1). The file is atomically updated to the second version by establishing the second inode as the inode for the file, wherein the establishing comprises storing the second inode in a journal (70, Fig. 4) stored in a nonvolatile memory (44, 48, Fig. 2) (see, e.g., specification page 15, line 28-page 16, line

1 and page 16, line 25-page 17, line 5). See generally, e.g., Figs. 2 and 4; specification page 9, line 26-page 10, line 14; page 15, line 14-page 17, line 24.

Appellant notes that, while various dependent claims are argued separately below, none of these dependent claims are in means plus function or step plus function form as permitted under 35 U.S.C. § 112, paragraph 6. Accordingly, since 37 C.F.R. § 41.37(v) only requires summary of the separately argued dependent claims that are subject to 35 U.S.C. § 112, paragraph 6, no summary of the dependent claims is provided in this section.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 2-3, 8, 12-15, 18, 22-23, 29-32, and 35 are rejected under 35 U.S.C. § 102(b) as being anticipated by Kozakura, U.S. Patent No. 5,724,581 ("Kozakura").
2. Claims 4-5, 9-10, 19-20, and 24-25 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Kozakura in view of Fuller, U.S. Patent No. 5,870,757 ("Fuller").
3. Claims 6-7, 16-17, 26-27, and 33-34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Kozakura in view of Zheng et al., U.S. Patent No. 6,571,259 ("Zheng").

VII. ARGUMENT

First Ground of Rejection:

Claims 2-3, 8, 12-15, 18, 22-23, 29-32, and 35 are rejected under 35 U.S.C. § 102(b) as being anticipated by Kozakura. Appellant traverses this rejection for at least the following reasons.

Claims 2, 12, 22, and 29:

Appellant respectfully submits that claims 2, 12, 22, and 29 recite combinations of features not taught or suggested in Kozakura. For example, claim 2 recites a combination of features including: "a non-volatile memory storing a first inode locating said first file in said storage and also storing a journal comprising a list of committed inodes; and a block manager … configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory, … record said second inode in said journal".

The Final Office Action mailed March 21, 2007 ("Office Action") alleges that the inodes are taught in Kozakura as the page tables. Appellant respectfully disagrees. Page tables locate physical pages stored in the memory system of the computer system, mapping logical pages used by the software to physical pages (see, e.g., Kozakura, col. 1, lines 32-28). **Page tables that *map logical pages to physical pages in memory* have**

nothing to do with inodes that *locate files on a storage*. Page tables cannot anticipate inodes, as they are completely different and are used for different purposes. Additionally, logical and physical pages are fixed in size (see, e.g., Kozakura col. 1, line 34), whereas files can have any size. Appellant notes that the standard for anticipation is one of fairly strict identity. Kozakura's page tables are significantly different from inodes, and cannot anticipate them.

Kozakura teaches current page tables that locate the latest physical page storing the latest update and the shadow physical page storing data before the latest update. The current page table data structures cannot be a journal, since they are updated as transactions progress and thus do not comprise a list of committed inodes. Kozakura also teaches a backup page table that is created at a checkpoint. However, the backup page table is also not a list of committed inodes. Page tables merely map logical pages to physical pages. Thus, a backup page table created at a checkpoint is merely a snapshot of the current memory state. There is no list. Furthermore, the current state at checkpoint is merely the state of the current and shadow pages. These pages may generally comprise any combination of committed and uncommitted data, and thus are not committed inodes.

Furthermore, Kozakura's **checkpoints are not related to a commit command**. Rather, they are performed cyclically or at a given time (Kozakura, col. 2, lines 48-49), or when no transactions are in progress (Kozakura, col. 2, lines 59-61). There is no command. Rather, Kozakura's system creates checkpoints automatically, without any command.

The Response to Arguments section of the Office Action responds to the checkpoint remarks above by stating that Kozakura obtains a currently unused physical page, copies data in the latest page table to the physical page, and enters the copied data in the management table 3 as the latest page table for the logical page. The copied-from latest page table is entered in the current page management table 3 as a shadow page table. Then, the newly-obtained physical page is set in the blank page management unit 6 as the physical page being used" (Kozakura, col. 5, lines 6-13) in response to "when data

on a logical page are updated as a result of a transaction" (Kozakura, col. 4, lines 66-67) (see Office Action, page 12, item 16). Appellant respectfully submits that the above highlighted teachings of Kozakura do not teach creating a checkpoint nor a shadow page table in response to a commit, either. Rather, the above teachings appear to refer to the creation of shadow page tables as updates are performed.

Claim 12 recites a combination of features including: "atomically updating said first file by establishing said second inode as the inode for said first file, wherein said establishing comprises storing said second inode in a journal stored in a nonvolatile memory". The same teachings of Kozakura highlighted above with respect to claim 2 are relied on to reject claim 12. Appellant respectfully submits that Kozakura does not anticipate the above highlighted features, either.

Claim 22 recites a combination of features including: "a non-volatile memory storing a first inode locating a first version of a file in said storage and also storing a journal comprising a list of committed inodes; and a block manager … configured to atomically update the file, producing a second version of the file, in response to a commit of the file by writing said second inode to said non-volatile memory… and wherein said block manager is configured to record said second inode in said journal". The same teachings of Kozakura highlighted above with respect to claim 2 are relied on to reject claim 22. Appellant respectfully submits that Kozakura does not anticipate the above highlighted features, either.

Claim 29 recites a combination of features including: "atomically updating the file to the second version by establishing said second inode as the inode for the file, wherein said establishing comprises storing said second inode in a journal stored in a nonvolatile memory". The same teachings of Kozakura highlighted above with respect to claim 2 are relied on to reject claim 29. Appellant respectfully submits that Kozakura does not anticipate the above highlighted features, either.

For at least the above stated reasons, Appellant submits that the rejection of

claims 2, 12, 22, and 29 is in error and requests reversal of the rejection. The rejection of claims 3-7 (dependent from claim 2), claims 13-20 (dependent from claim 12), claims 23-27 (dependent from claim 22), and claims 30-35 (dependent from claim 29) are similarly in error for at least the above stated reasons, and reversal of the rejection is requested. Each of claims 3-7, 13-20, 23-27, and 30-35 recite additional combinations of features not taught or suggested in the cited art.

Claim 8:

Appellant respectfully submits that claim 8 recites a combination of features not taught or suggested in the cited art. For example, claim 8 recites a combination of features including: "said first inode is stored in an inode file, and wherein said inode file is identified by a master inode, and wherein said inode file is atomically updated with said second inode by writing said master inode subsequent to said commit command".

The Office Action relies on the same teachings highlighted above with regard to claim 2 to allegedly teach the features of claim 8. No teachings of Kozakura cited in the rejection have anything to do with the above features. Furthermore, as noted above, the Office Action alleges that Kozakura's page tables correspond to inodes. However, there is no page table file, identified by a master page table, in Kozakura. Therefore, Kozakura cannot anticipate the above highlighted features of claim 8.

For at least the above stated reasons, Appellant submits that the rejection of claim 8 is in error and requests reversal of the rejection. The rejection of claims 9-10 (dependent from claim 8) is similarly in error for at least the above stated reasons, and reversal of the rejection is requested. Each of claims 9-10 recite additional combinations of features not taught or suggested in the cited art.

Claims 3 and 23:

Claims 3 and 23 depend from claims 2 and 22, respectively. Accordingly, the rejection of claims 3 and 23 is in error for at least the reasons highlighted above with regard to claims 2 and 22. Additionally, claims 3 and 23 recite combinations of features

including: "said commit of said first file comprises a commit command received from an external source which updates said first file."

The Office Action cites the same teachings of Kozakura highlighted above with regard to claim 2 to allegedly anticipate the above highlighted features. Appellant respectfully submits that there is no commit of a file, as noted previously, and thus there is no commit command received from an external source as recited above. Additionally, Kozakura does not teach or suggest an external source that updates the page tables. Accordingly, Kozakura does not teach or suggest the above highlighted features of claims 3 and 23.

For at least the above stated reasons, Appellant submits that the rejection of claims 3 and 23 is in error and requests reversal of the rejection.

Claims 18 and 35:

Claims 18 and 35 depend from claims 12 and 29, respectively. Accordingly, the rejection of claims 18 and 35 is in error for at least the reasons highlighted above with regard to claims 12 and 29. Additionally, claims 18 and 35 recite combinations of features including: "said establishing said second inode is performed in response to a commit command."

The Office Action cites the same teachings of Kozakura highlighted above with regard to claim 2 to allegedly anticipate the above highlighted features. Appellant respectfully submits that there is no commit command, as noted previously, and thus there is no establishing of the second inode in response to the commit command as recited above. Additionally, Kozakura does not teach or suggest inodes, and thus cannot establish the second inode (as the inode of the first file) in response to the commit command. Accordingly, Kozakura does not teach or suggest the above highlighted features of claims 18 and 35.

For at least the above stated reasons, Appellant submits that the rejection of

claims 18 and 35 is in error and requests reversal of the rejection.

Claims 13 and 30:

Claims 13 and 30 depend from claims 12 and 29, respectively. Accordingly, the rejection of claims 13 and 30 is in error for at least the reasons highlighted above with regard to claims 12 and 29. Additionally, each of claims 13 and 30 recite a combination of features including: "writing a master inode corresponding to an inode file including said second inode to a checkpoint record in said journal."

The Office Action cites the same teachings of Kozakura highlighted above with regard to claim 2 to allegedly anticipate the above highlighted features. Appellant respectfully submits that there is no inode file that includes the second inode, and no master inode corresponding to that inode file, in Kozakura. Accordingly, Kozakura does not teach or suggest the above highlighted features of claims 13 and 30.

For at least the above stated reasons, Appellant submits that the rejection of claims 13 and 30 is in error and requests reversal of the rejection.

Claims 14 and 31:

Claims 14 and 31 depend from claims 13 and 30, respectively. Accordingly, the rejection of claims 14 and 31 is in error for at least the reasons highlighted above with regard to claims 13 and 30. Additionally, each of claims 14 and 31 recite a combination of features including:

scanning said journal to locate a most recent checkpoint record and zero or more inodes subsequent to said most recent checkpoint record within said journal;

copying said master inode from said most recent checkpoint record to a volatile memory; and

updating an inode file corresponding to said master inode with said one or more inodes subsequent to said most recent checkpoint record.

The Office Action cites the same teachings of Kozakura highlighted above with regard to claim 2 to allegedly anticipate the above highlighted features. Appellant respectfully submits that there is no inode file, and no master inode corresponding to that inode file, in Kozakura. Accordingly, Kozakura does not teach or suggest the above highlighted features of claims 14 and 31.

For at least the above stated reasons, Appellant submits that the rejection of claims 14 and 31 is in error and requests reversal of the rejection.

Claims 15 and 32:

Claims 15 and 32 depend from claims 14 and 31, respectively. Accordingly, the rejection of claims 15 and 32 is in error for at least the reasons highlighted above with regard to claims 14 and 30. Additionally, each of claims 15 and 32 recite a combination of features including:

copying one or more blocks of said inode file storing said one or more inodes to a copied one or more blocks; and

updating said master inode in said volatile memory to point to said copied one or more blocks.

The Office Action cites the same teachings of Kozakura highlighted above with regard to claim 2 to allegedly anticipate the above highlighted features. Appellant respectfully submits that there is no inode file, and no master inode corresponding to that inode file, in Kozakura. Accordingly, Kozakura does not teach or suggest the above highlighted features of claims 15 and 32.

For at least the above stated reasons, Appellant submits that the rejection of claims 15 and 32 is in error and requests reversal of the rejection.

Second Ground of Rejection:

Claims 4-5, 9-10, 19-20, and 24-25 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Kozakura in view of Fuller. Appellant traverses this rejection for at least the following reasons.

Claims 4, 9, 19, and 24:

Claims 4, 9, 19, and 24 depend from claims 3, 8, 18, and 23, respectively. Accordingly, the rejection of claims 4, 9, 19, and 24 is in error for at least the reasons highlighted above with regard to claims 3, 8, 18, and 23. Furthermore, the addition of Fuller to the rejection does not cure the deficiencies in the rejection of claims 3, 8, 18, and 23. Additionally, each of claims 4, 9, 19, and 24 recite a combination of features including: "said commit command comprises a file close command".

The Office Action cites Fuller to allegedly teach the above features, admitting that Kozakura does not teach the above features. The Office Action cites the summary of Fuller and Table 1 in Fuller (col. 22, line 35-col. 23, line 23). Appellant respectfully submits that nothing in Fuller's summary teaches or suggests the above highlighted features. Additionally, Fuller describes Table 1 as follows: "Table 1 (hereinafter) illustrates 'commit' and 'NFS commit' assertions for various system calls. Fundamentally using the technique of the present invention, transacted operations will not cause synchronous writes if they do not require a commit and those transacted operations that do require a commit will generate fewer synchronous writes." (Fuller, col. 22, lines 16-21). Table 1 includes a system call that appears to be a file close ("TOP_CLOSE"). However, this call does not cause either type of commit operation to occur. (See the "Commit" and "NFS Commit" columns for TOP_CLOSE in Table 1). Accordingly, even if Kozakura and Fuller were combined as suggested in the Office Action, the combination would not teach or suggest "said commit command comprises a file close command."

For at least the above stated reasons, Appellant submits that the rejection of claims 4, 9, 19, and 24 is in error and requests reversal of the rejection.

Claims 5, 10, 20, and 25:

Claims 5, 10, 20, and 25 depend from claims 3, 8, 18, and 23, respectively. Accordingly, the rejection of claims 5, 10, 20, and 25 is in error for at least the reasons highlighted above with regard to claims 3, 8, 18, and 23. Furthermore, the addition of Fuller to the rejection does not cure the deficiencies in the rejection of claims 3, 8, 18, and 23. Additionally, each of claims 5, 10, 20, and 25 recite a combination of features including: "said commit command comprises an fsync command".

The Office Action cites Fuller to allegedly teach the above features, admitting that Kozakura does not teach the above features. The Office Action cites the summary of Fuller and Table 1 in Fuller (col. 22, line 35-col. 23, line 23). However, Appellant respectfully submits that it would not be obvious to modify Kozakura's page table management system with Fuller's teachings regarding a journaling file system. As highlighted above with regard to claim 2, page tables map logical pages to physical pages in a system memory, and have nothing to do with the file system on a disk. One would simply not be motivated to look to filesystem teachings when considering Kozakura's page table management system for system memory.

For at least the above stated reasons, Appellant submits that the rejection of claims 5, 10, 20, and 25 is in error and requests reversal of the rejection.

Third Ground of Rejection:

Claims 6-7, 16-17, 26-27, and 33-34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Kozakura in view of Zheng. Appellant traverses this rejection for at least the following reasons.

Claims 6 and 26:

Claims 6 and 26 depend from claims 2 and 22, respectively. Accordingly, the rejection of claims 6 and 26 is in error for at least the reasons highlighted above with regard to claims 2 and 22. Furthermore, the addition of Zheng to the rejection does not cure the deficiencies in the rejection of claims 2 and 22. Additionally, each of claims 6 and 26 recite a combination of features including: "said journal further includes a checkpoint record including a description of an inode file, a block allocation bitmap, and an inode allocation bitmap."

The Office Action alleges that the above highlighted features are taught in Zheng, specifically citing col. 3, line 3-col. 4, line 14 and col. 14, line 46-col. 15, line 14 of Zheng. Appellant respectfully submits that nothing in these sections teaches or suggests the above highlighted features. While the cited portions of Zheng do generally discuss block allocation and inode allocation, no mention of a checkpoint record, a block allocation bitmap, and an inode allocation bitmap is made.

For at least the above stated reasons, Appellant submits that the rejection of claims 6 and 26 is in error and requests reversal of the rejection.

Claims 7 and 27:

Claims 7 and 27 depend from claims 6 and 26, respectively. Accordingly, the rejection of claims 7 and 27 is in error for at least the reasons highlighted above with regard to claims 6 and 26. Additionally, each of claims 7 and 27 recite a combination of features including: "the description comprises inodes for each of said inode file, said block allocation bitmap, and said inode allocation bitmap."

The same sections of Zheng highlighted above with regard to claims 6 and 26 are cited to allegedly teach the features of claims 7 and 27. However, these sections do not teach or suggest the block allocation bitmap and the inode allocation bitmap, as highlighted above. Furthermore, Zheng does not teach or suggest inodes for the block allocation bitmap and the inode allocation bitmap, either.

For at least the above stated reasons, Appellant submits that the rejection of claims 7 and 27 is in error and requests reversal of the rejection.

Claims 16 and 33:

Claims 16 and 33 depend from claims 12 and 29, respectively. Accordingly, the rejection of claims 16 and 33 is in error for at least the reasons highlighted above with regard to claims 12 and 29. Furthermore, the addition of Zheng to the rejection does not cure the deficiencies in the rejection of claims 12 and 29. Additionally, each of claims 16 and 33 recite a combination of features including:

> said block map further comprises a first inode allocation bitmap indicating which inodes within said first inode file are allocated to files, the method further comprising:
>
> > copying said first inode allocation bitmap to a second inode allocation bitmap;
> >
> > modifying said second inode allocation bitmap to reflect one or more inodes allocated to new files; and
> >
> > establishing a third inode within said block map to said second inode allocation bitmap subsequent to said modifying said second inode bitmap.

The Office Action alleges that the above highlighted features are taught in Kozakura and Zheng, citing the same teachings of Kozakura and Zheng used in the rejections of the other claims. Appellant respectfully submits that nothing in these sections teaches or suggests the above highlighted features. While the cited portions of Zheng do generally discuss block allocation and inode allocation, no mention of a an

inode allocation bitmap is made, nor any copying of inode allocation bit maps and modifying of the copied bitmaps.

For at least the above stated reasons, Appellant submits that the rejection of claims 16 and 33 is in error and requests reversal of the rejection.

Claims 17 and 34:

Claims 17 and 34 depend from claims 16 and 33, respectively. Accordingly, the rejection of claims 17 and 34 is in error for at least the reasons highlighted above with regard to claims 16 and 33. Additionally, each of claims 17 and 34 recite a combination of features including:

> a first block allocation bitmap indicating which blocks within a storage including said block map are allocated to files, the method further comprising:
>
> copying said first block allocation bitmap to a second block allocation bitmap;
>
> modifying said second block allocation bitmap to reflect one or more blocks allocated to files; and
>
> establishing a fourth inode within said block map to said second block allocation bitmap subsequent to said modifying said second block allocation bitmap.

The Office Action alleges that the above highlighted features are taught in Kozakura and Zheng, citing the same teachings of Kozakura and Zheng used in the rejections of the other claims. Appellant respectfully submits that nothing in these sections teaches or suggests the above highlighted features. While the cited portions of Zheng do generally discuss block allocation and inode allocation, no mention of a an block allocation bitmap is made, nor any copying of block allocation bit maps and modifying of the copied bitmaps.

For at least the above stated reasons, Appellant submits that the rejection of

claims 17 and 34 is in error and requests reversal of the rejection.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejections of claims 2-10, 12-20, 22-27, and 29-35 are erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $510 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-59100/LJM.

Respectfully submitted,

/Lawrence J. Merkel/
Lawrence J. Merkel, Reg. No. 41,191
Agent for Appellant

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: January 17, 2008

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

2. A storage comprising:

a non-volatile memory storing a first inode locating a first file in said storage and also storing a journal comprising a list of committed inodes; and

a block manager configured to copy said first inode to a second inode, wherein said block manager is configured to change said second inode in response to updates to said first file, and wherein said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory, whereby said second inode locates said first file in said storage, and wherein said block manager is configured to record said second inode in said journal.

3. The storage as recited in claim 2 wherein said commit of said first file comprises a commit command received from an external source which updates said first file.

4. The storage as recited in claim 3 wherein said commit command comprises a file close command.

5. The storage as recited in claim 3 wherein said commit command comprises an fsync command.

6. The storage as recited in claim 2 wherein said journal further includes a checkpoint record including a description of an inode file, a block allocation bitmap, and an inode allocation bitmap.

7. The storage as recited in claim 6 wherein the description comprises inodes for each of

said inode file, said block allocation bitmap, and said inode allocation bitmap.

8. An apparatus comprising:

a computing node configured to perform one or more write commands to a file and a commit command committing the one or more write commands to said file; and

a storage coupled to receive said one or more write commands and said commit command, wherein said storage is configured to copy one or more blocks of said file to a copied one or more blocks, said one or more blocks updated by said one or more write commands, and wherein said storage is configured to update said copied one or more blocks with write data corresponding to said one or more write commands, and wherein said storage is configured to copy a first inode locating said file in said storage to a second inode and to update pointers within said second inode pointing to said one or more blocks to point to said copied one or more blocks, and wherein said storage is configured to atomically update said file by writing said second inode responsive to said commit command, and wherein said first inode is stored in an inode file, and wherein said inode file is identified by a master inode, and wherein said inode file is atomically updated with said second inode by writing said master inode subsequent to said commit command.

9. The apparatus as recited in claim 8 wherein said commit command comprises a file close command.

10. The apparatus as recited in claim 8 wherein said commit command comprises an fsync command.

12. A method comprising:

copying a first inode to a second inode, wherein said first inode locates a first file in a storage;

modifying said second inode in response to one or more changes to said first file; and

atomically updating said first file by establishing said second inode as the inode for said first file, wherein said establishing comprises storing said second inode in a journal stored in a nonvolatile memory.

13. The method as recited in claim 12 further comprising writing a master inode corresponding to an inode file including said second inode to a checkpoint record in said journal.

14. The method as recited in claim 13 wherein recovering from a system failure comprises:

scanning said journal to locate a most recent checkpoint record and zero or more inodes subsequent to said most recent checkpoint record within said journal;

copying said master inode from said most recent checkpoint record to a volatile memory; and

updating an inode file corresponding to said master inode with said one or more inodes subsequent to said most recent checkpoint record.

15. The method as recited in claim 14 wherein said updating said inode file comprises:

copying one or more blocks of said inode file storing said one or more inodes to a

copied one or more blocks; and

updating said master inode in said volatile memory to point to said copied one or more blocks.

16. The method as recited in claim 12 wherein said block map further comprises a first inode allocation bitmap indicating which inodes within said first inode file are allocated to files, the method further comprising:

copying said first inode allocation bitmap to a second inode allocation bitmap;

modifying said second inode allocation bitmap to reflect one or more inodes allocated to new files; and

establishing a third inode within said block map to said second inode allocation bitmap subsequent to said modifying said second inode bitmap.

17. The method as recited in claim 16 wherein said block map further comprises a first block allocation bitmap indicating which blocks within a storage including said block map are allocated to files, the method further comprising:

copying said first block allocation bitmap to a second block allocation bitmap;

modifying said second block allocation bitmap to reflect one or more blocks allocated to files; and

establishing a fourth inode within said block map to said second block allocation bitmap subsequent to said modifying said second block allocation bitmap.

18. The method as recited in claim 12 wherein said establishing said second inode is performed in response to a commit command.

19. The method as recited in claim 18 wherein said commit command is a close file command.

20. The method as recited in claim 18 wherein said commit command is an fsync command.

22. A storage comprising:

a non-volatile memory storing a first inode locating a first version of a file in said storage and also storing a journal comprising a list of committed inodes; and

a block manager configured to copy said first inode to a second inode, wherein said block manager is configured to change said second inode in response to updates to the file, and wherein said block manager is configured to atomically update the file, producing a second version of the file, in response to a commit of the file by writing said second inode to said non-volatile memory, wherein said second inode locates said second version of said file in said storage, and wherein said block manager is configured to record said second inode in said journal.

23. The storage as recited in claim 22 wherein said commit of the file comprises a commit command received from an external source which updates the file.

24. The storage as recited in claim 23 wherein said commit command comprises a file close command.

25. The storage as recited in claim 23 wherein said commit command comprises an fsync command.

26. The storage as recited in claim 22 wherein said journal further includes a checkpoint record including a description of an inode file, a block allocation bitmap, and an inode allocation bitmap.

27. The storage as recited in claim 26 wherein the description comprises inodes for each of said inode file, said block allocation bitmap, and said inode allocation bitmap.

29. A method comprising:

copying a first inode to a second inode, wherein said first inode locates a first version of a file in a storage;

modifying said second inode in response to one or more changes to the file, creating a second version of the file; and

atomically updating the file to the second version by establishing said second inode as the inode for the file, wherein said establishing comprises storing said second inode in a journal stored in a nonvolatile memory.

30. The method as recited in claim 29 further comprising writing a master inode corresponding to an inode file including said second inode to a checkpoint record in said journal.

31. The method as recited in claim 30 wherein recovering from a system failure comprises:

scanning said journal to locate a most recent checkpoint record and zero or more inodes subsequent to said most recent checkpoint record within said journal;

copying said master inode from said most recent checkpoint record to a volatile

memory; and

updating an inode file corresponding to said master inode with said one or more inodes subsequent to said most recent checkpoint record.

32. The method as recited in claim 31 wherein said updating said inode file comprises:

copying one or more blocks of said inode file storing said one or more inodes to a copied one or more blocks; and

updating said master inode in said volatile memory to point to said copied one or more blocks.

33. The method as recited in claim 29 wherein said block map further comprises a first inode allocation bitmap indicating which inodes within said first inode file are allocated to files, the method further comprising:

copying said first inode allocation bitmap to a second inode allocation bitmap;

modifying said second inode allocation bitmap to reflect one or more inodes allocated to new files; and

establishing a third inode within said block map to said second inode allocation bitmap subsequent to said modifying said second inode bitmap.

34. The method as recited in claim 33 wherein said block map further comprises a first block allocation bitmap indicating which blocks within a storage including said block map are allocated to files, the method further comprising:

copying said first block allocation bitmap to a second block allocation bitmap;

modifying said second block allocation bitmap to reflect one or more blocks allocated to files; and

establishing a fourth inode within said block map to said second block allocation bitmap subsequent to said modifying said second block allocation bitmap.

35. The method as recited in claim 29 wherein said establishing said second inode is performed in response to a commit command.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings known to Appellant.